

Korrespondance analyse af billeder

Christoffer Thrysøe, Andreas Borgstad

Indholdsfortegnelse

| | | |
|----------|---|-----------|
| 1 | Abstract | 1 |
| 2 | Introduktion | 2 |
| 2.1 | Opgavens problemfelt | 2 |
| 2.2 | Rapportens Opbygning | 3 |
| 2.3 | Problemformulering | 3 |
| 2.3.1 | Udvidelse af problemformuleringen | 3 |
| 2.3.2 | Afgrænsning | 3 |
| 3 | Korrespondanceanalyse | 4 |
| 3.0.3 | Kanter | 4 |
| 3.1 | Detektor | 6 |
| 3.1.1 | Hjørner | 7 |
| 3.1.2 | Blobs | 7 |
| 3.1.3 | Skalarum | 10 |
| 3.2 | Deskriptor | 12 |
| 3.3 | Matching | 13 |
| 4 | Analyse af markbilleder | 14 |
| 4.1 | Flyverute | 14 |
| 4.2 | Markbilleder | 14 |
| 4.3 | Hypoteser | 15 |
| 5 | Eksperimental Opsætning | 17 |
| 5.1 | Valg af metoder | 17 |
| 5.1.1 | Udvalgte detektorer | 17 |
| 5.1.2 | Udvalgte deskriptorer | 17 |
| 5.2 | Eksperimental opsætning | 17 |
| 6 | Gennemgang af udvalgte metoder | 18 |
| 6.1 | Moravec | 18 |
| 6.1.1 | Algoritme | 19 |
| 6.1.2 | Konklusion | 19 |
| 6.2 | Harris | 19 |
| 6.2.1 | Algoritme | 21 |
| 6.2.2 | Konklusion | 21 |
| 6.3 | SIFT | 21 |
| 6.3.1 | Detektion | 21 |
| 6.3.2 | Algoritme | 23 |
| 6.3.3 | Orientering af SIFT punkter | 24 |
| 6.3.4 | Deskriptor | 25 |
| 6.4 | SURF | 26 |
| 6.4.1 | Determinant of Hessian | 26 |
| 6.4.2 | Orientering | 28 |
| 6.4.3 | Deskriptor | 28 |
| 6.5 | Matching | 30 |
| | Litteratur | 32 |
| A | Appendix test | 33 |

1 Abstract

To Be Done

2 Introduktion

"Droner til monitorering af flerårigt ukrudt i korn" er et forsknings projekt, oprettet af Miljøstyrelsen, der i samarbejde med Datalogisk institut og institut for Plante- og Miljøvidenskab, har til formål at automatisere detektionen af ukrudt i marker. I dag bekæmpes ukrudt, ved at sprøjte hele marken med pesticider uanset, hvor ukrudtet er placeret. Projektet har til formål at nedsætte forbruget af pesticider, ved hjælp af en drone¹ udstyret med et kamera. Dronen tager et antal overlappende luftbilleder af en mark og disse billeder analyseres af en algoritme, der identificerer, hvor ukrudtet befinder sig i de enkelte billeder. Resultatet af denne analyse overføres til et globalt koordinatsystem af hele marken, hvilket kan opnås ved at bestemme de geometriske relationer imellem billederne. Det samlede ukrudtskort, vil angive ukrudtets placering og derved, hvor der er behov for pesticider [1].



Figur 1:

2.1 Opgavens problemfelt

Opgavens afsæt i projektet er etablering af korrespondancer, hvilket kan bruges til at estimere de geometriske relationer imellem billederne, igennem homografier. Estimering af de geometriske relationer imellem billederne tillader oprettelsen af ukrudtskortet. Følgende trin beskriver den process der er benyttet til at etablere korrespondancer mellem to billeder, hvilket i denne opgave betegnes som korrespondanceanalyse:

1. Interessante punkter udvælges i begge billeder, som resultat af at anvende en række matematiske modeller på billederne. Formålet ved dette stadie er at udvælge punkter i begge billeder, der repræsenterer projektionen af det samme fysiske objekt. Billederne overlapper i gennemsnit hinanden med ca. 70 %, hvilket gør korrespondanceanalysen mulig
2. For hvert interessepunkt oprettes en vektor, der beskriver den lokale billedstruktur omkring interessepunktet.
3. Beskrivelserne af interessepunkterne i de to billeder, sammenlignes for bestemme, hvilke punkter der korresponderer.

En nærmere definition af korrespondancebegrebet er givet i sektion 3.1.

¹UAV (eng. unmanned aerial vehicle)

2.2 Rapportens Opbygning

2.3 Problemformulering

Med udgangspunkt i litteraturen indenfor korrespondanceanalyse samt implementering af flere eksisterende metoder er spørgsmålet, hvilke metoder, der teoretisk og praktisk, bedst anvendes til korrespondanceanalyse af markbilleder.

2.3.1 Udvidelse af problemformuleringen

Der opstilles en beskrivelse af forskellige korrespondanceanalytiske metoder og foretages korrespondanceanalyse af markbillederne, ved implementering af disse metoder. Udvælgelsen af metoder er baseret på hypoteser om, hvad der kræves af metoderne, for at løse korrespondanceproblemet ved markbilleder (afsnit 4). Disse hypoteser vil eksperimentelt be- eller afkræftes, og derved give en empirisk forståelse for, hvilke forudsætninger metoderne skal besidde for at etablere korrespondancer. Det undersøges, hvilke metoder der muliggør en løsning af korrespondanceproblemet, samt en komparativ analyse af metodernes præstation.

2.3.2 Afgrænsning

Projektet sigter på at afprøve allerede eksisterende metoder på markbilleder og ikke at skabe nye metoder. Programmet konstrueres mht. afklaring af de nævnte problemstillinger og ikke mhp. efterfølgende at blive anvendt i praksis. De udvalgte metoder implementeres mhp. funktionaliteten. Derfor vil enkelte implementeringsdetaljer undlades, hvis formålet er at simplificere kompleksiteten.

3 Korrespondanceanalyse

3.0.3 metoder indenfor billedbehandling

Formelt kan et billede repræsenteres som en funktion af to variable:

$$\begin{aligned} I : \mathbb{R}^2 &\rightarrow \mathbb{R} \\ I(x, y) &= \lambda \quad (x, y) \in \mathbb{Z}^{+2}, \lambda \in [0, 1] \subseteq \mathbb{R} \end{aligned} \quad (1)$$

hvor λ repræsenterer en billedintensitet, også kaldet pixelværdi. λ er defineret, indenfor grænserne af billedet og er 0 udenfor, dvs.:

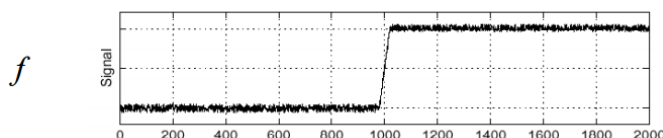
$$I(x, y) = \begin{cases} \lambda, & \text{hvis } 1 \leq x \leq x_{max}, 1 \leq y \leq y_{max} \\ 0, & \text{ellers} \end{cases}$$

En kort bemærkning bør gøres om det undersøgte billedformatet. De undersøgte billeder er af filtype JPEG og hver pixelintensitet indeholder originalt 8x3 bits information til hhv. rød, grøn og blå: $\lambda_{col} = [R, G, B]^T \in \mathbb{R}^3$. Hver farve kan antage $2^8 = 256$ værdier og hver værdi ligger i intervallet $[0, 1]$. Disse værdier bliver transformeret til en gråtone værdi, vha. Lumosity metoden <cite>:

$$Lum(\lambda_{col}) = [0.2126, 0.7152, 0.0722]\lambda_{col} = \lambda \quad (2)$$

Hver gang pixelværdi eller billedintensitet benævnes, er det underforstået, at de har undergået den lineære transformation, ligning [?]. x og/eller y kan højst antage værdien 65.535, og et billede er derfor i sagens natur diskret.

Diskontinuitet i billeder er ofte en nyttig egenskab, i en billedanalytisk procedure. Det kan være i form af en kant: Hvis et billede beskues i 3D, kan en kant illustreres i 1D, ved et snit af et billede vinkelret på overfladen, illustreret figur 2. En kant er en lokal ændring i det afledte signal f .



Figur 2: En 1-dimensional fortolkning af intensiteten i et billede. Intensitetsskiftet i midten ($x = 1000$, indikerer en kant).

En differentiering af funktionen fra figur 2 vil fremhæve dens udsving og derved angive hvor der forekommer kanter. Differentiering af billeder kan approsikmeres ved følgende ligning:

$$\frac{df(x)}{dx} = \frac{f(x+1) - f(x-1)}{2} \quad (3)$$

Foldning af I af størrelse $(M \times N)$, med en kerne K der har størrelse $(m \times n)$, hvor $M > m, N > n$:

$$O(i, j) = \sum_{x=1}^m \sum_{y=1}^n I(i+x-1, j+y-1) K(x, y) \quad (4)$$

Ligning (4) sker for alle $i, j \in I$. En kerne defineres her som en matrix af arbitrær dimension - ofte $(N \times N)$. En kerne foldes med et billede, for at opnå en effekt, f.eks. at glatte billedet, som er tilfældet med Gausskernen.

Et billede kan differentieres, som ligning (3), ved brug af foldning. Først defineres en kerne til horisontal differentiering K , som: $K = [\frac{1}{2}, 0, \frac{1}{2}]$. Kernen foldes med billedet I : $I * K$, hvor $*$

udgør foldningsoperatoren.

I grænsetilfælde, skal en kerne foldes med et billede, hvor afstanden til kanten af billedet, er skarpt mindre, end størrelsen af kernen - dvs. der iflg ligning (4) ganges med et område, hvor $x < 0, y < 0$. Her gælder:

Eg. $O(1, 1)$, med en kerne der har størrelse $M > m > 1, N > n > 1$, vil inputtet til I hvor $x + k - 1 < 0, l - 1 < 0$, være 0.

Det kan være problematisk at lokalisere kanter vha. differentiering. I figur 2, ses hvordan støj i billedet (de små udsving) også vil blive fremhævet, hvilket kan resultere i fejlagtige detektioner af kanter. For at fjerne støjen, kan billedet foldes med et Gaussisk filter, hvilket er en diskret approksimering til den Gaussiske funktion. Foldning af et billede med et Gaussisk filter vil resultere i en "flydende" overgang mellem pixelværdierne og derfor glatte billedet. Den Gaussiske funktion i 2-D, hvor σ er standardafvigelsen af den Gaussiske fordeling, er defineret som:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (5)$$

For at undgå først at glatte billedet ved at folde med et Gaussfilter, og derefter folde med et differentieringsfilter, udnyttes at foldning er en associativ operation:

$$\frac{\partial}{\partial x}(G * f) = (\frac{\partial}{\partial x}G) * f \quad (6)$$

Her er G Gausskernen, men kunne være en vilkårlig anden kerne, og f et signal.

Foldes et differentieret 1-dimensionelt Gaussfilter med signalet fra figur 2, vil det resultere i et bakkeformet signal, hvor bakken indikere en kant. For en mere lokaliserbar kant, kan den dobbelt afledte tages, som set i figur 3. I sidstnævnte tilfælde, kan kanten lokaliseres, hvor funktionen krydser nul.

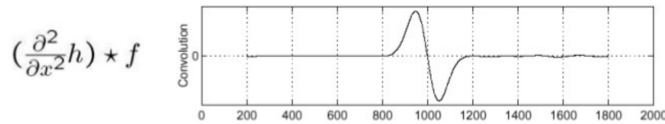
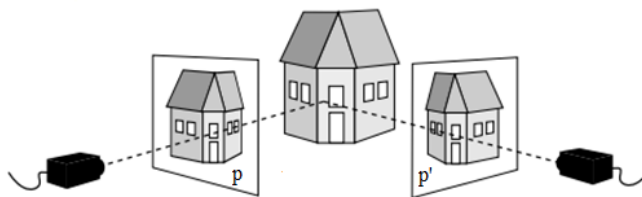


Figure 3: Resultatet af at folde et dobbelt differentieret Gaussisk filter med funktionen

I de metoder der i denne opgave er gjort brug af, er diskontinuiteter i lokale strukturer i billederne undersøgt. Derfor er det nyttigt at bruge et dataindsamlingsvindue \mathbf{D} . Dette er en (NxN) matrix af et udsnit af et billede. Givet to koordinater (x, y) , vil dataindsamlingsvinduet indeholde værdier der vil fremgå af konteksten, således, at $\mathbf{D}_{\frac{N}{2}, \frac{N}{2}} = f(x, y)$, hvor $f(x, y)$ er en funktion, der er tilknyttet koordinatet (x, y) i I - f.eks. billedintensiteten eller en udregnet gradient. Givet to billeder af samme scene, defineres korrespondanceproblemet ved at finde punkter i begge billeder, der er projektioner af samme fysiske scenepunkter. Korrespondanceproblemet løses derfor ved korrekt at udvælge sæt af punkter, der opfylder dette [12]. Udfordringen ved problemet er, at billederne, der skal etableres korrespondancer imellem er udsat for en række ændringer. Det omfatter forskydning af kameraets position, eller ændringer af scenens motiv, sket i tiden mellem to billeder er taget. Et eksempel på korrespondanceproblemet er det menneskelige syn: Øjnene agere som to kameraer, der indfanger, hver deres billede. Korrespondanceproblemet løses imellem de indfangede billeder, hvilket gør det muligt at opfatte dybde og bedømme afstand i en scene



Figur 4: Billeder af samme scene, fra to forskellige vinkler. p og p' angiver korresponderende punkter, da punkterne er de samme fysiske scenepunkter [2].

I figur 4 ses to billeder taget af samme scene, fra to forskellige vinkler. For at etablere korrespondancer, skal der udvælges punkter, som optræder i begge billeder, hvilket udføres af en detektor. I figur 4 er der detekteret to korresponderende punkter p og p' . Disse punkter, og den omkringliggende billedstruktur, skal beskrives af en deskriptor $Des(p)$. For en deskriptor skal det gælde, at: $Des(p) \approx Des(p')$, hvis p og p' repræsenterer korresponderende punkter. Endvidere skal der gælde, at $Des(p) \not\approx Des(p')$, hvis p og p' ikke repræsenterer korresponderende punkter. Korrespondanceanalyse består af følgende trin, som vil blive nærmere beskrevet i følgende sektion:

1. Feature Detektion
2. Feature Deskription
3. Feature Matching

3.1 Detektor

Feature detektion er en metode indenfor billedbehandling, der for hvert punkt $p = (x, y)$ i et billede I bestemmer, om punktet er et interessepunkt. Et interessepunkt er et punkt i et billede, der generelt kan karakteriseres ved[13]:

- *Definerbar struktur:* Punktet skal indgå i en struktur, der er matematisk definerbar. Dette er nødvendigt da punkterne skal kunne identificeres uafhængigt i begge billeder.
- *Repeterbar:* Repeterbarheden signficere uafhængighed af forskellige betingelser i billedet. Dvs. at samme interessepunkt skal kunne findes i to forskellige billeder, trods ændringer af f.eks lys og rotation
- *Informationsrigt:* Intensiteten i og omkring interessepunktet skal være unikt, dette muliggør en mere detaljeret beskrivelse af punktet. Et punkt placeret på et område af samme intensitet, vil ikke kunne differentieres ift. til andre punkter placeret i samme i område..

En feature detektor '*Detect*' er en funktion, der tager et billede som input, og returnere interessepunkter.

$$Detect(I) = \mathbf{P} \quad (7)$$

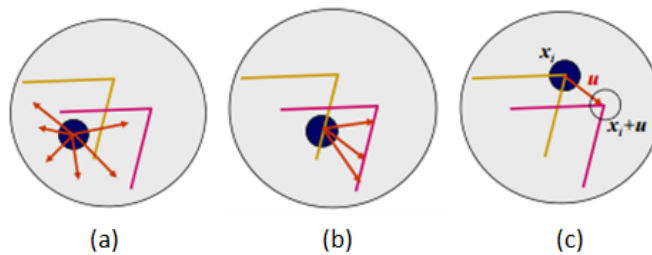
hvor $\mathbf{P} = p_1, p_2, \dots, p_n$ er et sæt punkter i billedet I udvalgt til at være interessepunkter. Moravec [10] definerer, som en af de første, et brugbart interessepunkt således: "A feature is good if it can be located unambiguously in different views of a scene. A uniformly colored region or a simple edge does not make for good features because its parts are indistinguishable. Regions, such as corners, with high contrast in orthogonal directions are best." «I følge Moravec, indeholder et godt interessepunkt en stor kontrast, i de ortogonale retninger». Der findes mange definitioner af et interessepunkt, og i sidste ende angiver anvendelsesområdet, hvilke punkter, der lokaliseres bedst. Interessepunkter defineres ikke udefra semantisk meningsfulde områder, som ansigter eller bøger, da dette vil kræve en høj-niveau fortolkning af scenen. I stedet anvendes lav-niveau strukturere, der identificeres i lokale pixelområder, der er matematisk definerbare. Nedenstående er en gennemgang af forskellige lokale lav-niveau strukturere, der kan bruges i udvælgelsen af interessepunkter.

3.1.1 Hjørner

At detektere hjørner er en udbredt teknik indenfor feature detektion, da hjørner ofte forekommer i forskellige menneskeskabte scener og fordelagtigt kan bruges i disse sammenhæng. Et hjørne kan defineres som et område i og omkring et punkt, der har to dominerende kantretninger og kan derved lokaliseres i et billede udefra en kvantificeret fortolkning af dette.

Udover at være lokaliserbar ved en matematisk definition, skal et interessepunkt være placeret således at det entydigt kan udvælges ift. dens omkringliggende område. I figur 5 ses tre udvalgte interessepunkter, placeret på forskellige strukturer i samme scene, hvor interessepunkterne beskrives af en blå cirkel omkring punktet. For at forstå, hvorfor punkter placeret på hjørner er gode interessepunkter, er mulige korrespondancer for de givne interessepunkter illustreret.

- Interessepunktet er placeret på et fladt område, hvor intensiteten er ens for hele området. Punktet vil have mange mulige korresponderende punkter.
- Interessepunktet er lokaliseret på en kant. Punktet vil have flere mulige korresponderende punkter på den tilsvarende kant i den anden figur.
- Interessepunktet er lokaliseret på et hjørne. Det ses at kun ét andet punkt i modsvarende scene er identisk med dette.

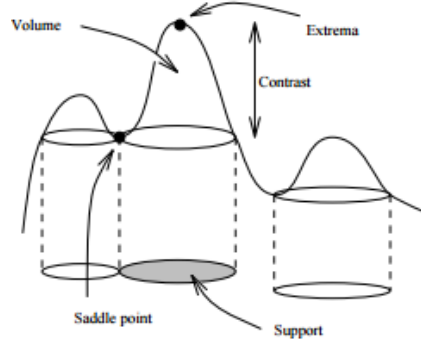


Figur 5:

Ovenstående definition viser at punkter placeret på hjørner er unikke og kan derfor entydigt differentieres fra ikke korresponderende punkter. Hjørner kan som nævnt udvælgelse i billederne pga. de dominerende kantretninger i og omkring punktet, hvilket medfører store intensitetsskift i området omkring et hjørne. En af de første hjørne detektorer udviklet af Moravec [?] definere et hjørne som værende et område med lav selvlighed, dvs. et område hvor der opstår store intensitetsvariationer i det omkringliggende område. Ud fra denne definition lokaliseres et hjørne ved at estimere auto-korrelationen imellem regioner af billedet, og definere hjørner, hvor der forekommer store intensitetsskift omkring et område.

3.1.2 Blobs

En blob er en region i et billede, hvor intensitet er konstant, og forskellig fra intensiteten udenfor regionen. Lindenberg [6] definere blobs som værende lyse regioner på sort baggrund eller omvendt - altså strukturer, der står i kontrast til deres baggrund.



Figur 6: En blob visualiseret i 2-d, udefra Lindenberg's definition [6]

I figur 6(a), ses en blob defineret af dens lokale ekstrema, hvor styrken af blobben beskrives ved kontrasten, ift. området omkring ekstremaet. Lindenberg definere en blob som værende afgrænset af dens saddelpunkt; et saddelpunkt angiver punktet, hvor intensiteten stopper med at falde og starter med at stige for lyse blobs, og modsat for mørke.

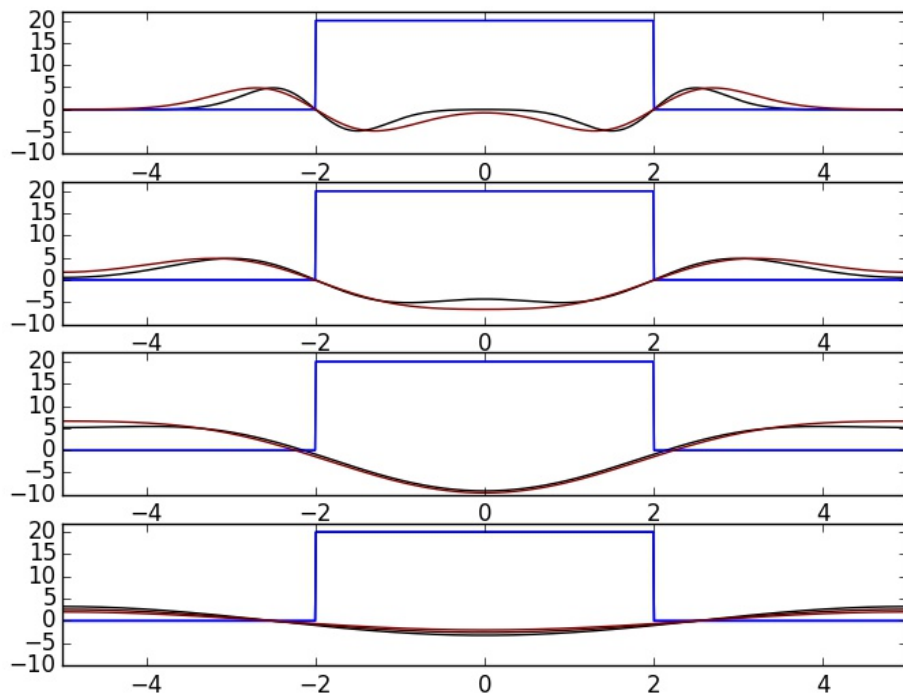
En metode til at detektere ekstremaer, er Laplace af Gauss (LoG). Laplace defineres således:

$$\Delta f = \nabla^2 f = \sum_{i=1}^n \frac{\partial^2 f}{\partial x_i^2} \quad (8)$$

Anvendes Laplace operatoren på Gaussfunktionen (ligning (5)), kan resultatet diskretiseres og anvendes som en kerne.

$$LoG = \sigma^2 \Delta G \quad (9)$$

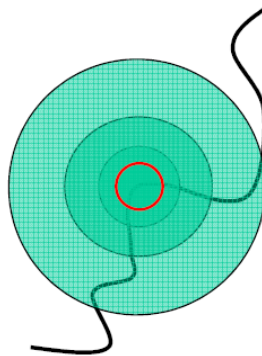
Bemærk, at σ^2 er blevet multipliceret på fra venstre. Dette er gjort for at normalisere LoG, således, at responset er invariant overfor størrelsen af σ . LoG kan diskritiseres og foldes med billedet for at finde lokale ekstremaer.



Figur 7: (a) En 3-D visualisering af en to-dimensional Laplacian of Gaussian (b) ét en-dimensionalt signal (c) Laplacian of Gaussian operatoren anvendt på (b)

Når blobs skal lokaliseres, skal σ værdien være tilpasses blobbens størrelse, som ses på figur (7). Figuren viser fire forskellige værdier for σ . I første illustration er σ værdien for lav - her dannes flere ekstremaer, men ingen af dem karakteriserer en blob, da den absolutte værdi for funktionen evalueret på det afledte af LoG signalet, er for lav. Dog ser kurven i illustration tre ud til, at have tilstrækkelig høj absolut værdi, til at kunne karakteriseres som et blob.

Problemet omkring valg af skala, kan afhjælpes ved skalarumsrepræsentation, som gennemgås i.



Figur 8:

I figur 8 angiver cirklerne forskellige undersøgte skalaer, Så hvordan udvælges cirklen, der dækker interesse området uafhængigt af områdets størrelse? For Blobs er det interessant når der i et skaleret område opstår et veldefineret ekstrema. En måde at søge efter ekstremaer over forskellige skalaer er ved at oprette et skalarum for det undersøgte billede, hvor hvert billede skales og der for hver skala findes interessepunkter. Skala-rummet i et 2-dimensionalt billede repræsenteres af flere billeder i forskellige skalaer af det originale billede. Billeder, der repræsenterer forskellige skalaer, opnås ved at folde billedet iterativt med et Gaussisk filter med stigende σ værdi.



Figur 9: Til venstre ses en visualisering af et skala-rum formet som en pyramide. Hvert niveau angiver en skala repræsentation af det originale vindue, hvor toppen af pyramiden indeholder billeder af største skala og derfor med mindst information, og bunden af skalaen med det originale billede. Til højre ses et billede foldet med et Gaussisk filter af stigende sigma værdier. Jo højere sigma værdi, jo flere fine detaljer bliver fjernet og billedet slørret.

Et Gaussisk filter bruges da gradvis højere værdier af σ fjerner fine strukturer, som vist i figur 9, og nye strukturer forekommer ikke ved transformationen fra finere til grovere skalaer [7]. Idéen er derved at fjerne disse strukturer og lede efter andre ekstremaer, gradvist på større skalaer, der også kan detekteres. Et billede i skalarummet for billedet $f(x, y)$ kan derfor defineres som i (10)

$$L(x, y, \sigma) = G(x, y, \sigma) * f(x, y) \quad (10)$$

hvor G er det 2-dimensionelle Gaussiske filter, $L(x, y, \sigma)$ repræsenterer et billede i skala-rummet, og skala-parametren σ , bestemmer skalaen, eller placeringen i skala-rummet. $L(x, y, 0) = f(x, y)$, da det er den "nederste" skala og den nederste del af pyramiden. Højere niveauer af pyramiden kan opnås ved at folde billedet med et Gaussisk filter af større sigma værdi.

3.1.3 Skalarum

En Detektor skal være robust overfor små geometriske og fotometriske deformationer samt støj. Geometriske forskelle kan omfatte rotation, skalering og ændring af perspektiv. Fotometriske forskelle kan omfatte ændringer i belysning (f.eks. ved delvist overskyet vejr) eller forskelle i refleksion mod de to drone-positioner hvorfra billederne er optaget. En god detektor skal være invariant eller robust over for sådanne forskelle. Flere artikler understreger nødvendigheden for skalainvarians hos detektorer og at problemet skal konfronteres i alle billedbehandlings situationer [?] [6] [7]. Denne sektion beskriver, hvordan en sådan invarians kan opnås.

Objekter i virkeligheden, såvel som detaljer i et billede, optræder kun som meningsfulde enheder over et specifikt skala-interval. Et træ vil indenfor centimeters eller nanometers afstand optræde som blade eller molekyler og indenfor meters afstand, vil opfattes som et træ. Koenderink [?] definere et objekts skala-interval som værende afgrænset af to skalaer: den inderste og den yderste skala, F.eks. vil en trætop have en indre skala på ca. 10 cm og en ydre skala på ca. 10 m.

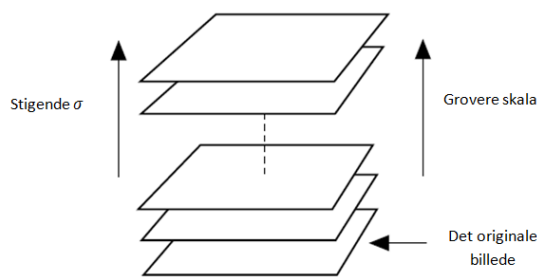
Formelt kan et billede repræsenteres som en funktion af to variable:

$$\begin{aligned} I : R^2 &\rightarrow R \\ I(x, y) &= \lambda \quad (x, y) \in R^2, \lambda \in R \end{aligned} \quad (11)$$

hvor λ repræsenterer en billedintensitet. Da billedet indeholder en ukendt scene, vides det ikke hvilket skala-interval, der for et givent objekt indgående i scenen, er meningsfuldt. Den inderste skala af billedet vil derfor være bundet af pixelstørrelsen og den ydre af billedets fysiske størrelse. En aksiomatisk tilgang til dette problem er at undersøge en bred skalarepræsentation af billedet i form af en udvidelse af billedfunktionen med en enkelt parameter også kaldt skalaparametren σ :

$$\begin{aligned} L : R^3 &\rightarrow R \\ L(x, y, \sigma) &= \lambda \quad (x, y, \sigma) \in R^3, \lambda \in R \end{aligned} \quad (12)$$

For at opnå en multi-skala repræsentation af billedet, oprettes der et skalarum bestående af skala-billeder, der går fra at udtrykke finere til grovere strukturer, proportionelt med skalaparametren, som illustreret i figur 10.

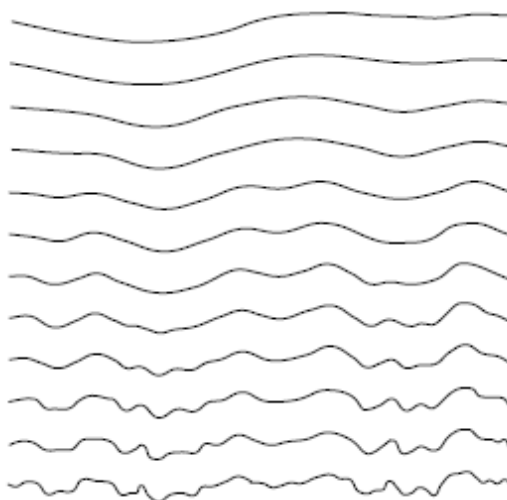


Figur 10:

Denne overgang fra finere til grovere strukturer kan opnås ved iterativt at folde billederne med et Gaussisk filter af stigende σ værdi, hvor billedets nulskala repræsenteres ved billedet $L(x, y, 0) = I(x, y)$ og for $\sigma > 0$:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (13)$$

Der anvendes et Gaussisk filter, på grund af dens unikke egenskaber, bl.a. at der ikke forekommer nye strukturer ved glatning, når skalaparametren stiger. Glatningen af billedet kan ses som at billedet flades ud og derved vil intensiteten i maksima falde og stige i minima. Witkins [?] beviste dette i tilfældet for én-dimensionelle signaler illustreret i figur 13, der viser resultatet af at glatte et signal med et Gaussisk filter af stigende σ værdi. Det ses tydeligt, hvordan signalets underliggende grovere struktur bliver udtrykt og at finere strukturer undertrykkes.



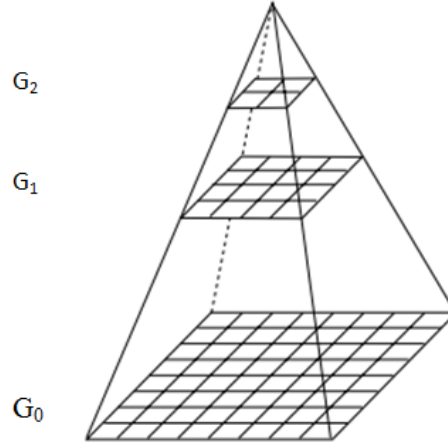
Figur 11:

Koenderink [?] førte dette begreb videre til det to-dimensionelle tilfælde, ved at vise at foldningen af billedet med et Gaussisk filter er den generelle løsning til diffusionsligningen, som beskriver hvordan en varmfordeling I udvikler sig over tid t .

$$\partial_t I = \nabla^2 I$$

Skala Pyramide

En måde, at repræsentere et skalarum er ved en *skalapyramide*. En skalapyramide repræsentation er en datatype, der opretter en række kopier af et billede, der iterativt reduceres i størrelse. Imellem, hvert niveau af pyramiden er billedet foldet med et Gaussisk filter og derefter halveret i størrelsen.



Figur 12:

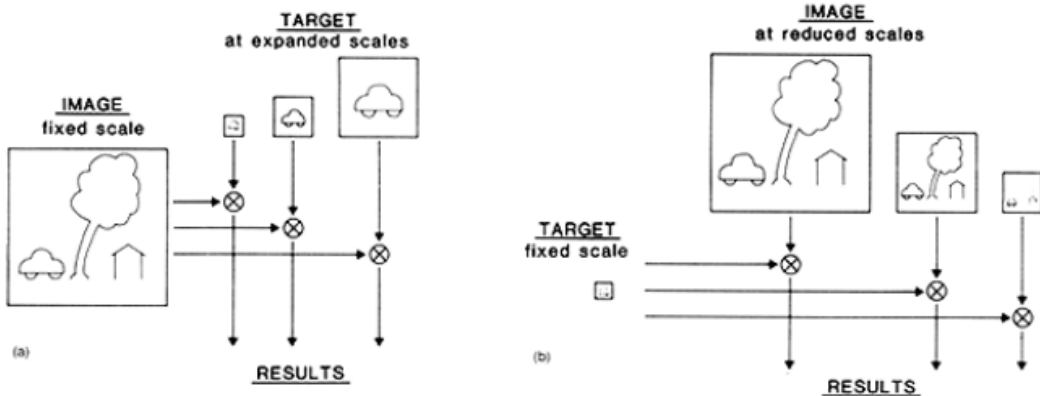
Hvis *REDUCE* betegner operatoren der reducere billedet og G er skalapyramide repræsentation, hvor G_0 er det originale billede, kan de forskellige niveauer opnås ved:

$$G_l = REDUCE[G_{l-1}] \quad (14)$$

Nærmere kan operationen beskrives som:

$$G_l(i, j) = \sum_m \sum_n w(m, n) G_{l-1}(2i + m, 2j + n) \quad (15)$$

hvor w er et Gaussisk filter. Figur 13 viser sammenhængen med at (a) anvende et Gaussisk filter af iterativt stigende størrelse, og (b) ved at oprette en skalapyramide.



Figur 13:

Fordelene ved en pyramide repræsentation er at billedernes størrelser reduceres, hvilket reducere antallet af beregninger drastisk.

3.2 Deskriptor

Fra en detektor genereres et sæt interessepunkter \mathbf{P} , fra to eller flere billeder som vist i ligning (7). For at udvælge korresponderende punkter imellem billederne, skal interessepunkterne beskrives af en deskriptor Des . Deskriptoren er en funktion, der tager et billede og et punkt som input. Deskriptoren beskriver dette punkt, udfra den lokale billedstruktur, omkring og i punktet og tilegner punktet en beskrivelse, kaldet en feature f :

$$Des(I, p_i) = f_i$$

$$Des(I', p'_j) = f'_j$$

For to korresponderende punkter i og j ønskes det at $f_i \approx f'_j$, så punkterne korrekt kan identificeres som værende korresponderende punkter, men også for to *ikke* korresponderende punkter k og l at $f_k \not\approx f'_l$ så de kan identificeres som ikke korresponderende. Den tilegnede beskrivelse kommer i form af en n -dimensional vektor², hvor indgangene i vektoren indeholder information om området omkring interessepunktet, hvilket kan bruges til at determinere, ligheden af interessepunkterne.

$$f = \begin{bmatrix} d_1 \\ d_2 \\ \dots \\ d_n \end{bmatrix}$$

Generelt ønskes der at en deskriptor besidder følgende egenskaber:

- *Robust*: Deskriptoren skal kunne identificere to korresponderende punkter som værende ens på trods af ændringer i billedet.
- *Invariant*: Deskriptoren skal være invariant overfor de ændringer, der kan forekomme under fotograferingen.

3.3 Matching

Fra deskriptoren er et sæt interessepunkter \mathbf{P} beskrevet ved et sæt af features \mathbf{F} . En matching funktion *Match* tager disse to sæt (\mathbf{P}, \mathbf{F}) , hver bestående af (p_i, f_i) . Funktionen tager disse to sæt af punkter med tilsvarende features, og returnerer et sæt parvise punkter vurderet til at korrespondere. Matching funktionen kan beskrives som:

$$Match(\mathbb{P}, \mathbb{P}') = \mathbf{M} \tag{16}$$

Hvor alle indgange i \mathbf{M} består af parvise koordinater, e.g. (p_i, p'_j) . Feature vektorene i det ene billede skal sammenlignes med feature vektorer i det andet billede. Pga. ændringer i billederne vil korresponderende features aldrig være helt identiske, og der skal derfor oprettes nogle metoder, der kan determinere, hvornår to Features er ens nok til at tilsvarende punkter korrespondere.

²For en given metode vil længden af disse vektorer være ens

4 Analyse af markbilleder

Denne sektion har til formål at klarlægge udfordringer ved korrespondanceanalysen af markbilleder. Udfordringerne opstilles på baggrund af en analyse af billederne og vil resultere i hypoteser om, hvilke forudsætninger de korrespondanceanalytiske metoder skal besidde for at muliggøre en korrespondanceanalyse.

4.1 Flyverute

Dronens overflyvning foregår ved at dronen får fire punkter, der definerer markens placering. Dronen starter derefter i et hjørne og flyver systematisk fra side til side, med samme orientering, for at overflyve hele marken. Dronen starter på en højde af ca. 60 meter, hvorefter den foretager en overflyvning, falder i højde og flyver marken over igen. I det udvalgte billedsæt er der fire overflyvninger, hvor hver overflyvning foregår på nogenlunde samme højde.

4.2 Markbilleder

I denne opgave er et sæt billeder fra samme overfløjede kornmark anvendt. På billedsættet ses en kornmark, der består af umiddelbare tilfældige kornstrukturer. I enkelte billeder er der traktorspor. Som illustreret i figur <Indsæt billede der viser udsnit af korn>, er kornet af relativ identisk udseende, og det er derfor vanskeligt at differentiere imellem korn områder, da der ikke opstår definerbare mønstre. Ligheden imellem billederne kan derfor besværliggøre korrespondanceanalysen, hvilket gør det interessant at undersøge hvorvidt det er muligt at opnå korrespondancer. figur 14.

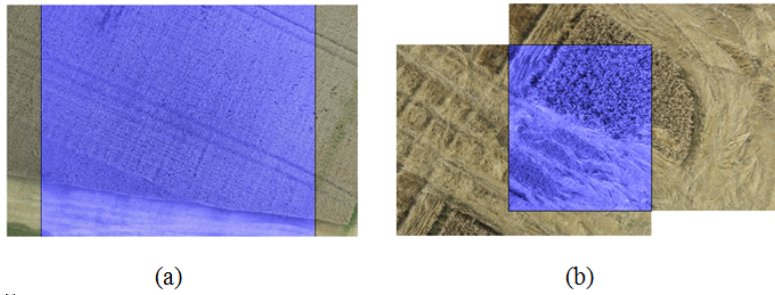


Figur 14: *Markbillederne forstørret ind på enkelte korn.*

Følgende observationer er gjort af markbillederne, samt forhold omkring overflyvningen:

- **Overlap:** Overlappet imellem billederne har en signifikant indflydelse på korrespondanceanalysen. Et stort overlap vil betyde at interessepunkter har større sandsynlighed for at indgå i begge billeder. Derfor, ved mindre overlap, skal der udvælges flere interessepunkter. Det er bekræftet at billederne overlapper hinanden med ca. 70%³, der forekommer dog store udsving i overlappet, alt efter hvilken højde billederne er taget på. Figur 15 viser fire billeder, sat sammen parvist, der er taget ved to forskellige højder, hvor billedernes overlap er markeret med blå. (a) er taget når dronen er på største højde, billederne er estimeret til at overlappe hinanden med ca. 80 %. (b) er taget ved en lavere højde, billederne overlapper hinanden med ca. 35 %.

³JOHN



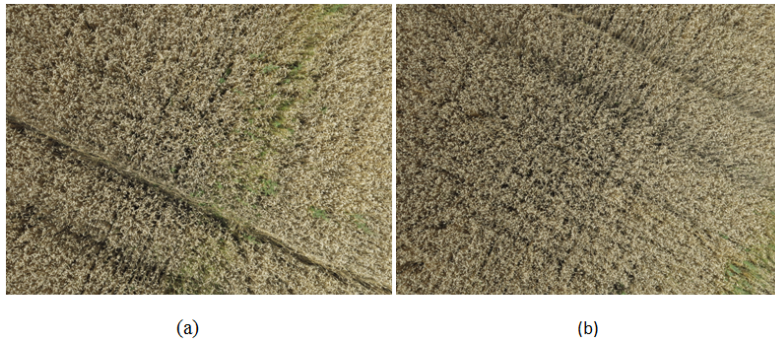
Figur 15: Markbilleder taget fra forskellige højde, det blå område viser billedernes overlap.

- **Rotation:** Der er observeret mindre grad af rotation imellem billederne. Når dronen er nået kanten af marken er det registreret at den i enkelte tilfælde rotere med op til 10-15 °.



Figur 16: Dronen skal til at ændre retning, hvilket giver rotation i billederne.

- **Okklusion:** Korn, der stikker direkte op imod kameraet, ses som mørke områder pga. den synlige jord, hvilket vil resultere i et mørkere område direkte under dronen. Forskydes kameraets position, vil det samme område nu ligne korn, da jorden ikke længere kan ses.



Figur 17: To forkudte billeder, hvor traktorsporet er okkluderet.

Figur 17 er et eksempel på okklusion af jord, hvor traktor spor optræder direkte under dronen og derefter forskudt. Traktorsporet, der optræder direkte under dronen, er tydelig og man se jorden. For det samme traktorspor, i det forskudte billede, er jorden ikke længere synlig. Det er også værd at bemærke at kornet, direkte under dronen, afgiver mørke områder. Denne okklusion forekommer umiddelbart kun når dronen er på lav højde.

- **Struktur**

4.3 Hypoteser

Ud fra ovenstående analyse er der opstillet følgende hypoteser om krav til de udvalgte metoder:

- <Det forventes, grundet strukturen af korn, at detektorer som leder efter veldefinerede strukturer, som hjørner, ikke vil have en god effekt på korn, men at blobs vil give bedre resultater.>
- Det forventes ikke at metoderne behøver være rotationsinvariante, grundet den lille rotation der forekommer imellem billederne.
- Det forventes at metoderne skal være skalainvariante. Bla. for at tilpasse størrelsen af kornet, <men også pga. flere artikler, der strengt understreger nødvendigheden for skalainvarians >.
- Grundet det til tider lille overlap imellem billeder, forventes det at detektoren skal tilpasses til at finde mange punkter, for at sikre en repeterbar detektion.
- Det forventes ikke at der skal tages højde for problemer ved okklusion, da det kun foregår når dronen flyver lavest. Et overvejet tiltag var ikke at undersøge et område direkte under dronen, men det blev konkluderet at eventuelle nye interessepunkter, der opstår ved okklusion vil sorteres væk som ikke korrespondancer.

5 Eksperimental Opsætning

Denne sektion har til formål..

5.1 Valg af metoder

Metoderne er først og fremmest udvalgt mhp. at opnå en succesfuld korrespondanceanalyse af markbilleder. De udvalgte metoder er derfor anerkendte og veldokumenterede. Derudover er metoderne udvalgt mhp. at be - eller afkræfte de opstillede hypoteser omkring, hvilke nødvendige forudsætninger metoderne skal besidde for at opnå en succesfuld korrespondanceanalyse af markbilleder.

5.1.1 Udvalgte detektorer

Følgende detektionsmetoder er implementeret:

- Moravec[10]. En hjørnedetektor introduceret af H. Moravec i 1980.
- Harris[?]. En hjørnedetektor introduceret af C.Harris & M.Stephens i 1988.
- Difference of Gaussian[8]. En skala invariant blobdetektor, introduceret af D.Lowe i 2004.
- Determinant of Hessian [3]. En skala invariant blobdetektor, introduceret af H.Bay, T.Tuytelaars & L. Van Gool. i 2006

Detektionsmetoderne er udvalgt for at undersøge nødvendigheden af skalainvarians og, hvilken struktur, der detekteres bedst i markbilleder.

5.1.2 Udvalgte deskriptorer

- SIFT [8]. En deskriptor introduceret af D. Lowe i 2004
- SURF [3]. En deskriptor introduceret af H.Bay, T.Tuytelaars & L. Van Gool. i 2006
- U-SURF⁴

Deskriptorene er udvalgt for undersøge nødvendigheden af rotationsinvarians.

5.2 Eksperimental opsætning

<hvilke billeder brugt> <Kombinationer>

⁴Upright-SURF, en ikke rotationsinvariant version af SURF

6 Gennemgang af udvalgte metoder

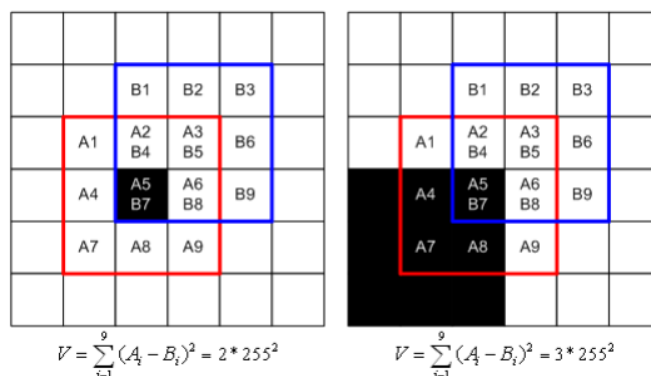
I denne sektion vil implementerede metoder gennemgås og eventuelle implementerings afvigelser blive nævnt. Til sidst vil en overordnet konklusion, diskutere hver metodes styrke og svaghed.

6.1 Moravec

Moravecs hjørne detektor[10], er en af de tidligste feature detektorer, og definere et hjørne som værende et område med lav selvlighed. Moravec anvender denne definition til at opnå en matematisk formulering af hjørner ved, for hvert punkt i billedet, at udregne auto-korrelationen af overlappende pixel regioner af 3x3, 5x5 eller 7x7 dataindsamlingsvinduer. En høj auto-korrelation imellem de overlappende pixelregioner vil betyde lav selvlighed og derved et hjørne. Dataindsamlingsvinduet kan derved bruges til at detektere hjørner, ved at forskyde dette med én pixel i alle principielle retninger (horisontalt, vertikalt og diagonalt) og udregne intensitetsvariationen (auto-korrelationen) af alle de forskudte vinduer. Denne variation kan beskrives med en vægtet funktion over vinduet (*SSD: Summed Square Difference*), der udregner forskellen på vinduerne:

$$E(u, v) = \sum_{\forall x, y \text{ i dataindsamlingsvinduet}} w(x, y) [I(x + u, y + v) - I(x, y)]^2 \quad (17)$$

hvor $(u, v) \in \{-1, 0, 1\}$ er forskydningen af vinduet, der summere over alle pixels i dataindsamlingsvinduet w , der har centrum i (x, y) . Figur 18 illustrere en udregning af *SSD* af et diagonalt skift på en isoleret sort pixel (med en intensitet på 0) på en hvid baggrund (med en intensitet på 255) og på et ideelt hjørne. Det røde vindue indikere det originale vindue og det blå indikere vinduet forskudt med vektoren i retningen (1, 1).



Figur 18: *SSD* udregninger for intensitetsvariationer imellem forskudte vindue.

Af de otte skift genereres otte intensitetsvariationer $E(u, v)$ ift. det originale vindue. Den mindste forskel af disse otte variationer, definere punktets *hjørnestyrke* $C(x, y)$.

$$C(x, y) = \min(E(u, v))$$

En horisontal kant vil have en stor selvlighed (lav auto-korrelation) med et dataindsamlingsvindue forskudt horisontalt derfor, for at identificere hjørner, tages minimum af E for at sikre en lav selvlighed (stor auto-korrelation) i alle retninger. Der opstilles en grænseværdi t for C , der bestemmer om punktet er et hjørne(sandt/falsk) og derved om det kan udvælges som et interessepunkt.

$$\text{hjørne} = \begin{cases} \text{sandt} & \text{hvis } C(x, y) \geq t, \\ \text{falsk} & \text{hvis } C(x, y) < t. \end{cases} \quad (18)$$

Moravec lider af følgende problemer pga. dens simplicitet:

- Der undersøges kun et diskret sæt af pixelskift (i hver principiel retning) og resultatet er derfor anisotropisk, (afhængig af retning). Undersøges en kant, der ikke er horisontal, vertikal eller diagonal, vil den mindste intensitetsvariation være stor, og derved kan punktet fejlagtigt detekteres som et hjørne.
- Det skiftende vindue er rektangulært, og metoden er derfor meget følsom overfor støj i billedet.
- Detektoren finder punkter lokaliseret på kanter. Små deformationer i kanterne som støj, vil resultere i at den mindste intensitetsvariation vil være relativt stor, og derfor detektere punktet som værende et interessepunkt.

6.1.1 Algoritme

1. For hvert pixel i billedet, udregn auto-korrelationen imellem skift af $(u, v) \in \{-1, 0, 1\}$. udregnet ved ligning 17
2. Find "Hjørnestyrken" ved at tage $C(x, y) = \min(E(u, v))$
3. Fjern punkter der ikke overholder en sat grænseværdi for $C(x, y)$.

6.1.2 Konklusion

Moravec er som nævnt en simpel algoritme, med mange udfordringer, der gør at den ikke bruges som en repeterbar detektor. Detektoren er i dag ikke i sig selv relevant, som den var da den blev udgivet, men bygges videre på i andre detektorer, f.eks. [5] beskrevet i sektion, 6.2 som direkte tilgår de nævnte problemstillinger med Moravec.

6.2 Harris

Harris hjørnedetektor blev introduceret af Chris Harris og Mike Stephens i 1988 [?]. Metoden refereres til som Harris hjørnedetektor, men er også kendt som Plessey detektor. Harris hjørnedetektor kan ses som en videreudvikling af Morave, ved at adressere Moravecs begrænsninger:

1. *Anisotropisk respons*: Harris og Stephens udvider Moravecs udregning af auto-korrelation til at måle intensitetsvariationer i alle retninger. Auto-korrelationen imellem skiftende vinduer kan approksimeres ved at udvide funktionen, ved en "Taylor udvidelse" af anden orden, og beskrives derved udefra dens afledte:

$$E(u, v) = \sum w(x, y) [I(x + u, y + v) - I(x, y)]^2 \quad (19a)$$

$$\approx \sum w(x, y) [I(x, y) + uI_x + vI_y - I(x, y)]^2 \quad (19b)$$

$$= \sum w(x, y) [u^2 I_x^2 + 2uv I_x I_y + v^2 I_y^2] \quad (19c)$$

Ligning (19c) kan omskrives til matrixform

$$E(u, v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} \quad (20)$$

Hvor M er auto-korrelations matricen (også kaldt struktur-tensoren), med $w(x, y)$ ganget på:

$$M = w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (21)$$

Billedets afledte i x og y akser findes ved at udføre en diskret differentiering af billedet, hvilket kan opnås ved at folde billedet, med henholdsvis $[-1 \ 0 \ 1]$ og $[-1 \ 0 \ 1]^T$.

2. *Støj sensitiv*: Harris detektoren anvender et Gaussisk vindue $w(x, y)$ omkring det undersøgte pixel område. Dette tilføjer en vægt i form af en cirkel omkring punktet. Det Gaussiske vindue anvendes til at fjerne støj fra billedet, da differentiering er følsom overfor dette.
3. *Højt kant respons* Struktur-tensoren 'M' opskrevet i (??) indeholder de differentieret retninger i et givet punkt og derved en beskrivelse af billedets geometriske struktur, i et punkt (x, y) . Egenverdierne for denne matrix, vil være proportionale med den principielle krumning i billedets flade og er derfor interessant ift. at finde hjørner. Egenverdierne vil være proportionale med de forskellige steder vinduet kan placeres: <overvej autokorrelations billeder fra springer>
 - a) *Homogen flade*. Billedintensiteten er konstant. Begge egenverdier vil være små og der vil ikke være nogen krumning i billedet.
 - b) *Kant*. For punkter placeret på en kant, vil der være en stor krumning på tværs af kanten, men ingen når kanten følges. Derfor vil én egenverdi være stor, mens den anden vil være lille
 - c) *Hjørne*. Et punkt placeret på et hjørne, vil have en stor krumning, i hver retning og derved bestå af to store egenverdier.

Det ønskes derfor, for at identificere hjørner, at begge egenverdier for M er store. Egenverdierne kan defineres ud fra determinanten og sporet af M:

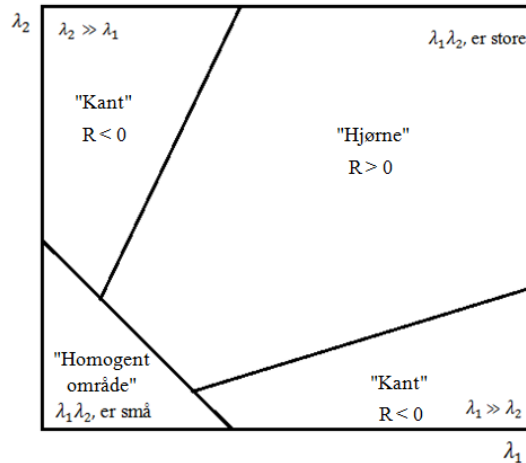
$$\det M = \lambda_1 \lambda_2 \quad (22a)$$

$$\text{trace} M = \lambda_1 + \lambda_2 \quad (22b)$$

Harris og Stephens foreslår et mål for hjørnestyrken baseret på forholdet imellem de to egenverdier. Hvor en høj værdi angiver to store egenverdier. Dette defineres således:

$$R = \det M - k(\text{trace} M)^2 \quad (23)$$

hvor k er en empirisk defineret konstant mellem 0.04 – 0.06. I figur 19 ses sammenhængen mellem værdien R , egenverdierne og hvordan de korresponderer med hjørner, kanter og homogene områder, igen kan det ses at for et hjørne ønskes to store egenverdier.



Figur 19: Opdeling af egenverdi rummet til specifikke regioner.

Ved værdien for R , kan det defineres om et punkt er lokaliseret på et hjørne. De interessante hjørne, er hvor der opstår en stor "hjørnestyrke", hvilket kan findes ved at sætte en grænseværdi t , der definerer om et hjørne er lokaliseret (sandt/falsk).

$$\text{hjørne} = \begin{cases} \text{sandt} & \text{hvis } R \geq t, \\ \text{falsk} & \text{hvis } R < t. \end{cases} \quad (24)$$

6.2.1 Algoritme

1. For hvert pixel i billedet (x, y) , udregn struktur tensoren, beskrevet i ligning (21), hvor $I_x^2 = (\frac{\partial I}{\partial x})^2 * w$, og w er et Gaussisk vindue.
2. Konstruer et billede bestående af hjørnestyrken 'R' udregnet ved ligning (23)
3. Opstil en grænseværdi for billedet bestående af 'hjørnestyrken', som i ligning (24)

6.2.2 Konklusion

Metoden anvendes som en repeterbar detektor og bruges i høj grad i dag. Detektoren er rotationsinvariant pga. dens brug af egenværdier, men er ifølge [9] stadig delvis anisotropisk, hvilket skyldes differentierings filtrene, der kun udledes i x og y retningen. Mikolajczyk og Schmid [9] konkludere at detektoren er særlig følsom over for rotation omkring 45 grader. De konkludere også at detektorens repeterbarhed falder drastisk ved skalaændringer.

6.3 SIFT

SIFT⁵ er en korrespondanceanalyse metode, introduceret af David Lowe i 2004 [8]. SIFT er en samlet løsning, der består af en detektor og en deskriptor.

6.3.1 Detektion

SIFT gør brug af en metode kaldt Difference of Gaussian(DoG), til at finde interessepunkter. DoG er en skala-invariant blob detektor, og er en approksimation til en skalanormaliseret Laplacian of Gaussian(LoG). DoG kan udregnes, ved en subtraktion af et nærliggende skalabillede, separeret af en konstant k , hvor G er et Gaussisk filter.

$$\begin{aligned} DoG(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \quad (25)$$

Den skalanormaliseret Laplacian of Gaussian kan opstilles som:

$$\begin{aligned} LoG(x, y, \sigma) &= \sigma^2 \nabla^2 L(x, y, \sigma) \\ &= \sigma^2 (L_{xx} + L_{yy}) \end{aligned} \quad (26)$$

Lowe relatere approksimeringen af DoG til LoG, ved diffusions ligningen:

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 L \quad (27)$$

Ligning (27) kan approksimeres til:

$$\sigma \nabla^2 L \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \quad (28)$$

Omskrivning af ligning (28) giver:

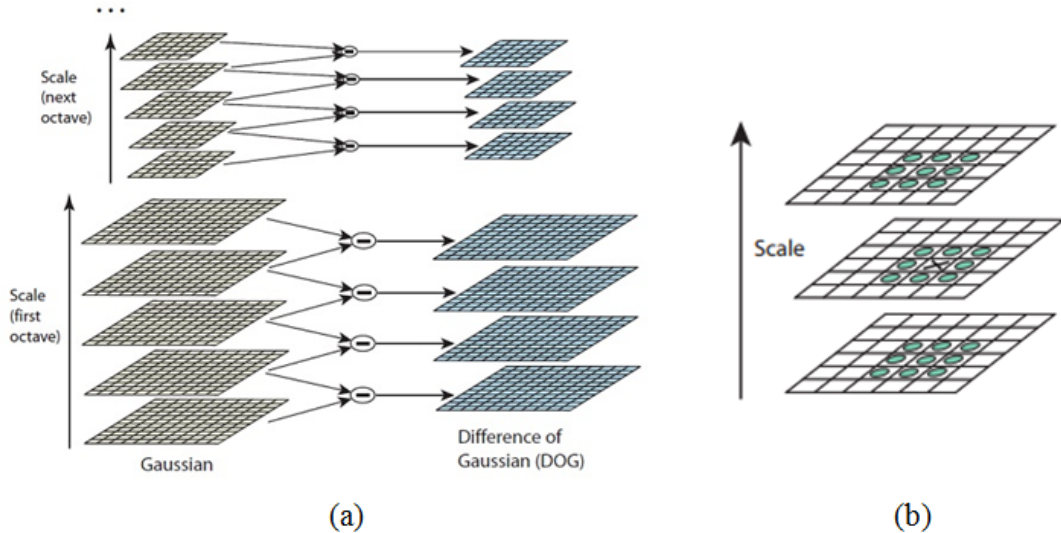
$$\begin{aligned} (k\sigma - \sigma) \sigma \nabla^2 L &\approx L(x, y, k\sigma) - L(x, y, \sigma) \\ (k - 1) \sigma^2 LoG &\approx DoG \end{aligned} \quad (29)$$

Metoden anvender et Skalarum, der deles op i oktaver, hver bestående af skalabilleder. Skalabilleder opnås ved at folde billedet iterativt med et Gaussisk filter, af stigende sigma værdi, dvs. med

⁵Distinctive Image Features from Scale-Invariant Keypoints

en stigende grad af slørring. Sigma værdien for første skalabillede, for en given oktav l er dobbelt så stor som forrige: $\text{Start}\sigma_l = 2 \cdot \text{Start}\sigma_{l-1}$. Sigma værdien for næste skalabillede opnås ved at multiplicere sigma værdien med en konstant k , dvs. $\sigma_n = \sigma_{n-1} \cdot k$. I den udførte implementering er $k = \sqrt{2}$, hvor den første sigma værdi er 1.6. Lowe anbefaler at bruge fire oktaver, hver med syv skalabilleder. I denne implementering er der brugt fire oktaver, med fem skalabilleder per oktav, hvilket har vist sig, at give brugbare resultater. Metoden anvender en skalapyramide og billedet reduceres derfor, for hver oktav, med halv størrelse, ift. forrige oktav.

Når skalarummet er oprettet skal DoG billederne produceres, hvilket er illustreret i figur 20(a), ved at subtrahere et skalabillede i en given oktav, med forrige skalabillede i samme oktav.



Figur 20: (a) Nærliggende skalabilleder for hver oktav trækkes fra og producerer DoG billederne. (b) I DoG billederne udvælges ekstremaer blandt 27 nabo pixels.

LoG svarer til sporret af hessian matricen som angiver summen af hessian matricens egenverdier: $\nabla^2 L = \text{tr}(\mathcal{H}) = \lambda_1 + \lambda_2$. Hessian matricens egenverdier beskriver den principielle krumning i og omkring et punkt og kan derfor bruges til at udvælge forekomster af blobs. Der vil opstå et maksima når $\nabla^2 L < 0$, da egenverdierne begge vil være negative og et minima når $\nabla^2 L > 0$ da egenverdierne vil være positive. Der ønskes derved at identificere ekstremaer i DoG billederne, da de vil angive forekomster af blobs. For hver pixel, sammenlignes et $3 \times 3 \times 3$ område, som vist figur 20 (b). Hver pixel bliver sammenlignet med dens $8 + 9 + 9 = 26$ naboer i de omkringliggende DoG billeder. Et punkt udvælges som værende et ekstrema, hvis det er det største/mindste iblandt de 26 naboer.

Når lokale ekstremaer er udvalgt fravælges dårligt lokaliserede punkter, f.eks. ekstremaer med lav kontrast, men også punkter, hvor ekstremaer ligger imellem pixels. Ligeledes bliver punkters placering fundet, med subpixel nøjagtighed. Dette gøres, ved non-maximal suppression, som foreslået af Lowe [11]:

$$D(x) = D + \frac{\partial D^T}{\partial x} x \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \quad (30)$$

Ovenstående kan forstås om en Taylor udvidelse af punktet D. Lokationen af ekstremaet \hat{x} findes ved at tage den afledte af den ovenstående funktion ift. x og sætte den til 0:

$$\hat{x} = \frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \quad (31)$$

I denne implementering er Non-maximal suppression anvendt anderledes ift. den anvendt i SIFT: Lowe foreslår, at hvis et punkt er et ekstrema, som udregnet ved ligning 30 og $\hat{x} > 0.5$ i en retning, skal ligning (30) udregnes omkring det nye punkt og dette gentages, indtil $\hat{x} < 0.5$, eller

indtil udregning er udført et max antal gange.

I denne implementering bliver et punkt fjernet, hvis $\hat{x} > 0.5$ i en given retning, da dette kun vil øge antallet af interessepunkter, hvilket er vurderet til ikke at være nødvendigt, da der i forvejen lokaliseres et stort antal punkter. Det vil ligeledes øge kompleksiteten af applikationen.

Punkter med lav kontrast fjernes, ved at sætte grænseværdi, for $D(\hat{x})$:

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \quad (32)$$

Lowe foreslår at værdier af $D(\hat{x})$ under 0.03 fjernes. Metoden vil også have en positiv respons overfor kanter, derfor anvendes metoder lånt fra Harris og Stephens [?] til at fjerne disse. Hessian matricen opstilles som:

$$\mathcal{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (33)$$

En negativ determinant angiver et saddle-point og fjernes derved. For at fjerne punkter lokaliseret på en kant beskrives forholdet imellem hessian matricens egenverdier, ved følgende ligning, hvor r angiver størrelsesforholdet imellem egenverdierne:

$$\frac{\text{tr}(\mathcal{H})^2}{\text{Det}(\mathcal{H})} < \frac{(r+1)^2}{r} \quad (34)$$

En grænseværdi r kan opstilles for at fjerne punkter lokaliseret på en kant. Punkter der tilfredsstiller denne grænseværdi udvælges som interessepunkter.

6.3.2 Algoritme

Input:

Output:

1. Konstruer skalarummet for de forskellige oktaver, ved iterativt at folde skalabilledet med en stigende værdi af σ :

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

og halver størrelsen af billedet for hver oktav.

2. Udregn "*Difference of Gaussian*"billeder, ved at tage forskellen imellem skala billederne, som i ligning (25).
3. Lokaliser ekstremaer for hvert punkt i "*Difference of Gaussian*"billeder, ved at sammenligne punktet med dens 27 naboer som illustreret i figur 20.
4. Afvis punkter der ikke er lokaliseret på ekstremaer, og ekstremaer med lav kontrast som i ligning (31) og (32):

$$\text{indikator} = \begin{cases} \text{fjern} & \text{hvis } \hat{x} > 0.5 \vee |D(\hat{x})| > 0.03 \vee \det \mathcal{H} < 0, \\ \text{behold} & \text{ellers.} \end{cases} \quad (35)$$

5. Fjern punkter lokaliseret på en kant ved at opstille en grænseværdi r og fjern punkter der ikke opfylder denne, som i ligning (34).

6.3.3 Orientering af SIFT punkter

Orienteringen af punkter anvendes i deskriptoren, for at opnå invarians overfor rotation. Skridtet her, kan beskrives ved:

$$\text{Orientering}(p, \sigma) = \theta \quad (36)$$

Hvor p er interessepunktet, σ er skalaen, og θ er orienteringen af punktet.

Et 16×16 dataindsamlingsvindue er placeret omkring interessepunktet, på skalabilledet som interessepunktet er fundet på. For alle punkter i dataindsamlingsvinduet, er en størrelsen på en gradient, og dens orientering beregnet:

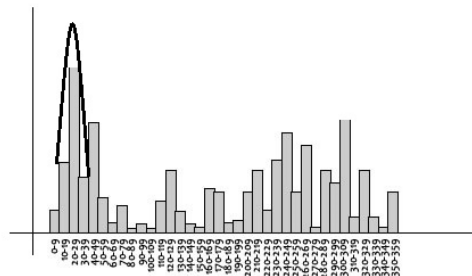
$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (37)$$

$$o(x, y) = \tan^{-1}((L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2) \quad (38)$$

m er størrelsen af en gradient, og o er gradientens retning. For alle punkter indenfor dataindsamlingsvinduet, skal (37) og (38) udregnes, for at skabe et gradientvindue g og orienteringsvindue v , begge vinduer af størrelse 16×16 . g skal herefter foldes med et Gaussisk filter hvor $\sigma_{Gauss} = 1.5\sigma_{point}$, med størrelse 16×16 .

Der skal oprettes et orienteringshistogram H , med 36 indgange. Alle gradienter med orientering v skal tilføjes, vægtet af gradientstyrken g , til H . En indgang i H , dækker en vinkel på 10° . F.eks. skal alle gradienter med vinkler mellem $0^\circ - 10^\circ$, tilføjes vægtet, til H_1 , osv. Dette resulterer i et orienteringshistogram, hvor indgangen i histogrammet, med størst værdi, bliver bearbejdet. Lowe foreslår, at alle indgange i histogrammet, der ligger indenfor 80% af det højeste punkt, bliver nye features - dette er undladt her, for at undgå at få for mange matches.

Der skal nu foretages en interpolation, omkring den indgang i histogrammet, med størst værdi, for at få et mere præcist estimat af θ .



Figur 21: Histogrammet H afbilledet, sammen med en andengradsligning. Andengradsligningen er beregnet over den største indgang i H

Dette ses på figur 21: En andengradsligning er blevet tilnærmet den største værdi af H . Estimering af andengradspolynomiet sker over den største værdi i H , og dennes umiddelbare venstre og højre nabo.

Algoritme

1. Dataindsamlingsvindue placeres omkring interessepunkt, på samme skala og variant af billedet, som σ foreskriver.
2. (37), (38) bruge til at udregne alle størrelser og orienteringer af gradienter, g , v , respektivt. g foldes, så: $g = G(x, y, 1.5\sigma)$.

3. Værdierne i g adderes på indgange i histogrammet H , afhængigt af deres tilsvarende orientering i v .
4. En interpolation laves omkring den største værdi i H og dens tætteste naboer. Denne værdi returneres

6.3.4 Deskriptor

SIFT deskriptoren skaber invarians overfor skala, belysning og vinkel. Der benyttes et dataindsamlingsvindue W , der har størrelsen 16×16 , til at indsamle information omkring et interessepunkt, og denne information udtrykkes som en vektor, med 128 indgange. Deskriptoren kan formelt beskrives:

$$Des(I, p, \sigma, \theta) = F \quad (39)$$

Hvor I er billedet, p er et interessepunkt (x, y) , σ er den tilhørende skalaparameter punktet er fundet på, og θ er orienteringen af p . F er en feature vektor, tilegnet punktet p .

Skalaen for et givent punkt bestemmer, om det billede der skal foretages beregninger på, skal være subsamplet. For at opnå rotationsinvarians, bliver dataindsamlingsvinduet W roteret i forhold til θ . Dette opnås, ved at tage prik produktet, af hver indgang i dataindsamlingsvinduet, med rotationsmatricen:

$$W_{mn_{new}} = W_{mn} \cdot \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \quad (40)$$

$W_{mn_{new}}$ bruges til at indsamle information om størrelsen og orienteringen af gradienten i det i 'ende punkt, som i (37) og (38) respektivt - gradienten og orienteringen bliver gemt i to separate matricer af størrelse 16×16 , hhv $Grad$ og Ori . $Grad$ bliver herefter foldet med et Gaussisk filter størrelse 16×16 , hvor sigmaværdien er halv så stor, som størrelsen af vinduet ($= 8$):

$$Grad_{new} = G(x, y, 8) * Grad(x, y) \quad (41)$$

Herefter skal G_{new} inddeles i 16 regioner ($R_{ij} \subseteq (G_{new})$), hver med størrelse 4×4 , som set i figur(???). Der skal foretages trilineær interpolation, og alle de kontinuerte orienteringer i O , skal vægtes som beskrevet nedenfor, og værdierne i figur (???, b), der svarer til F_i , skal opdateres - her er F_i repræsenteret ved 16 vinduer, hver med 8 retninger (svarede til 45 grader). Dette sker i 3 skridt:

1. alle punkter i R_{ij} skal vægtes efter, hvor langt de ligger fra centrum af R_{ij} . Det er vedtaget, at der eksisterer tre længder (beskriv v. billede!!!)
2. 4 indgange i F_i opdateres efter punkterne i R_{ij} 's orientering. Punkterne i R_{ij} skal fordeles på de 4 af de 8 orienteringer, der ligger tættest orienteringen af R_{ij} . Disse skal vægtes med G_{new} og værdien fundet i 1.
3. De 4 indgange, der er blevet opdateret, skal tilsvarende opdateres i 3 naboområder, af F_i (hvis der eksisterer nogen)

F normaliseres, og alle værdier større end 0.2 skal sættes til 0.2, og F skal normaliseres igen.

Algoritme

1. Det bestemmes via σ hvilket billede der skal udføres beregninger på
2. W roteres, ved rotationsmatricen (40), og glattes med (41)
3. R_{ij} oprettes, ved at dele $Grad_{new}$ op i 16 regioner, hver med størrelse 4×4
4. Alle punkter, i alle regioner af R_{ij} , skal nu bruges til at opdatere F , som beskrevet de 3 skridt ovenfor

6.4 SURF

SURF⁶, introduceret af Herbert Bay et. al [3] i 2006 er, ligesom SIFT, en samlet metode, bestående af en detektor og en deskriptor, der finder og beskriver blobs af forskellige størrelser. I forhold til SIFT, har SURF den fordel, at den beregningsmæssigt kan optimeres til at være betydeligt hurtigere end SIFT.

6.4.1 Determinant of Hessian

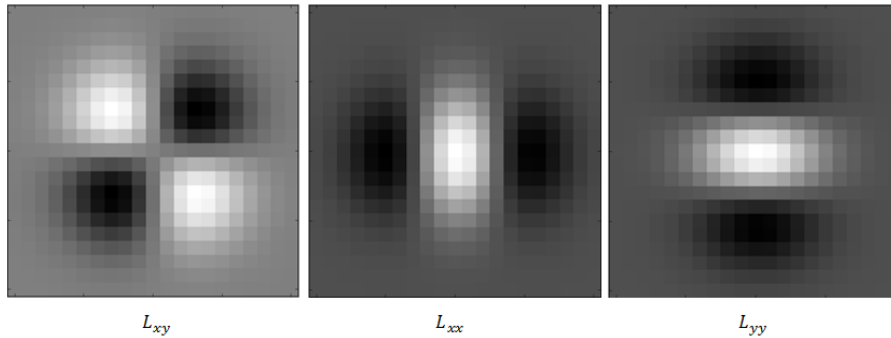
Determinant of Hessian eller *DoH* udgør feature detektoren i SURF. DoH er baseret på Hessian matricen, som udregnes for hvert punkt p i et billede:

$$\mathcal{H}(p, \sigma) = \begin{bmatrix} L_{xx}(p, \sigma) & L_{xy}(p, \sigma) \\ L_{xy}(p, \sigma) & L_{yy}(p, \sigma) \end{bmatrix} \quad (42)$$

hvor σ angiver skala og L_{xy} , L_{yy} og L_{xx} (vist i ligning(43)), er den Gaussiske funktionen, partielt differentieret ift. xy , yy og xx .

$$L_{xx}(x, \sigma) = \left(\frac{\partial^2}{\partial x^2} G(x, y, \sigma) \right) * I \quad (43)$$

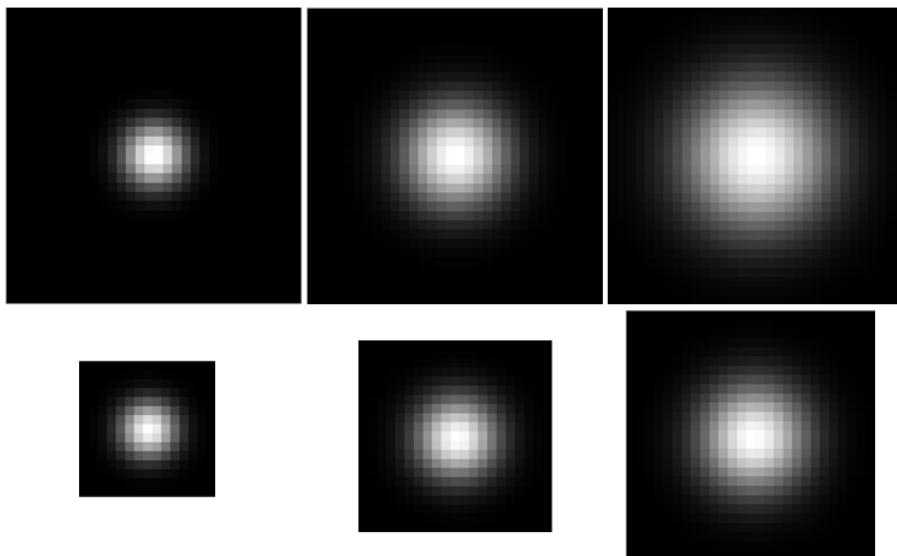
Bay et al. anvender approkimerede boks filtre: D_{xx} , D_{yy} og D_{xy} , der udelukkende består af værdierne $-2, -1, 0, 1$ som fordelagtigt, ved brug af integralbilleder, nedsætter antallet af beregninger drastisk. I denne implementering er disse boks filtre ikke anvendt, og er derved udregnet som ligning (43). Figur 22 er en illustration af de afledte Gaussiske filtre, der er anvendt.



Figur 22:

I SURF oprettes et skalarum, der ligesom i SIFT opdeles i skalaer og oktaver. I stedet for at reducere billedernes størrelser, forøges størrelserne af de anvendte filtre. SIFT anvender en stigende sigma værdi, med en fast filterstørrelse. Dette er det samme som at anvende en stigende filterstørrelse med en fast sigma værdi. I figur 23 ses forholdet imellem størrelserne af den Gaussiske funktion.

⁶SpeededUp Robust Features



Figur 23: Øverst ses den gaussiske funktion, hvor der ligesom i SIFT er vedligeholdt en filterstørrelse, men med stigende sigma værdi. Nederst ses den samme funktion, der ligesom i SURF vedligeholder den samme sigma værdi, men anvender stigende filterstørrelser.

For hver oktav anvendes der, i denne implementering, fire billeder foldet med fire forskellige filterstørrelser. For hver oktav, bruges nr. 2 filterstørrelse, fra forrige oktav til starten af næste, og størrelsen imellem filtre, fordobles fra forrige oktav, som vist i tabel 24. Her skal det bemærkes, at de andenafledte filtre, skal have dobbelt størrelse, af de førsteafledte.

| oktav | filter str 1 | filter str 2 | filter str 3 | filter str 4 |
|-------|--------------|--------------|--------------|--------------|
| 1 | 9 | 15 | 21 | 27 |
| 2 | 15 | 27 | 39 | 51 |
| 3 | 27 | 51 | 75 | 99 |
| 4 | 51 | 99 | 147 | 195 |

Figur 24: Fire forskellige oktaver, og filterstørrelse, for et andenafledt filter

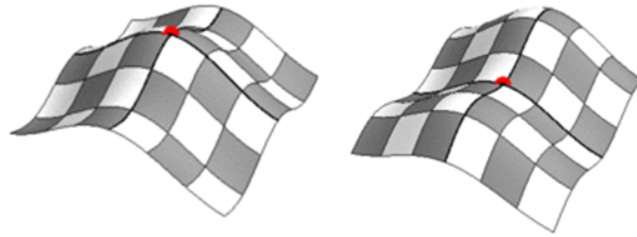
For hvert punkt i billederne opstilles Hessian matricen, hvorefter determinanten udregnes for alle punkter i billederne. Bay et.al. udregner determinanten ved:

$$\det \mathcal{H}_{\text{approximeret}} = D_{xx}D_{yy} - wD_{xy}^2 \quad (44)$$

hvor w er en vægt, tilføjet for at balancere de approksimerede differentierede Gaussiske filtre og boks filtrene. Da denne implementering ikke anvender boks filtre, udregnes determinanten som:

$$\det \mathcal{H} = L_{xx}L_{yy} - L_{xy}^2 \quad (45)$$

Hessian matricen anvendes, da den beskriver den lokale geometriske krumning omkring et punkt. Hessian matricens egenvektorer angiver de principielle retninger af krumningerne, og egenværdierne beskriver størrelserne af disse krumninger. For blobs ønskes et lokalt ekstrema, og derved at egenværdierne enten begge er positive eller begge er negative. Er dette ikke gældende vil blobben have en krumning i hver sin retning og derved være et saddle-point. Determinanten udregnes, da det er produktet af egenværdierne og derfor fortæller noget om fortegnet på disse. En positiv determinant vil betyde at egenværdierne er af samme fortegn, hvor en negativ vil betyde at fortegnet for egenværdierne er forskellige og punktet derfor placeret på et saddle-point. I figur 25(venstre) ses et punkt placeret på et maxima, hvor krumningen er ens i alle retningerne derfor vil egenværdierne være negative og determinanten positiv. I det andet billede (højre) ses et punkt placeret på et saddle-point. Krumningerne er forskellige, hvilket vil resultere i egenværdier af forskellige fortegn og derved en negativ determinant.



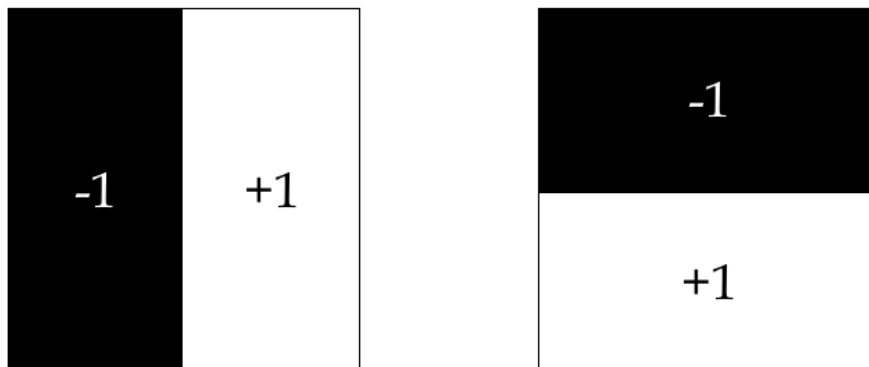
Figur 25: Til venstre ses et punkt placeret på et maksima. Til højre ses et punkt placeret på et saddle-point

Metoden reagere derved på mørke og lyse blobs, ved positive determinanter. Derfor udvælges et lokalt maxima af et $3 \times 3 \times 3$ område af determinant billederne, som vist i figur 20. Dette udføres for alle oktaver. Herefter udvælges korrekt subpixel placering, hvilket er en metode introduceret af David Lowe, også beskrevet i SIFT. Igen udføres dette skridt ved at fjerne punkter, der ikke er lokaliseret tæt nok på ekstremaer.

Algoritme

1. Hessian matricen opstilles for alle punkter i skalabillederne som i ligning (42)
2. Determinantbilleder opstilles, ved at udregne determinanten af Hessian matricen for alle punkter i alle billeder som i ligning (45).
3. Lokale maxima udvælges, af hvert $3 \times 3 \times 3$ område af billeder på samme oktav.
4. Accurate keypoint localisation, bruges til at fjerne dårligt lokaliserede punkter ligesom i ligning (35).

6.4.2 Orientering



Figur 26:

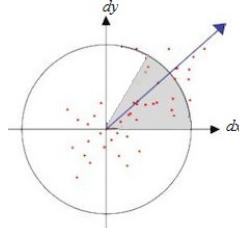
SURF gør brug af Haar-wavelet, både i detektoren og deskriptoren. Haar-wavelet er et boks-filter, med størrelse $n \times n$, hvor halvdelen af alle indgangene er $+1$ og den anden halvdel er -1 .

6.4.3 Deskriptor

SURF deskriptoren producerer features, der består af 64 indgange. Den kan beskrives stringent, ligesom (26). Som beskrevet i SURF papiret af Tuytelaars et. al[??], er tilfældet ofte, at der ikke er brug for rotationsinvarians. En variant af SURF, der ikke er rotationsinvariant, kaldes Upright-SURF (U-SURF). Tuytelaars et. al garanterer dog rotationsinvarians i U-SURF, i op til $\pm 15^\circ$. SURF er implementeret, fremfor U-SURF, da enkelte af billederne har stor grad (ca. 15°) af rotation. Rotationen forekommer i billeder, der er taget lige efter, at dronen har vendt og kan

ses på figur 15.

For at gøre SURF invariant overfor rotation, skal Haar-wavelet responset findes i x og y retningen omkring punktet (dx, dy) , respektivt. dx og dy beregnes i en cirkel omkring interessepunktet, med radius 6σ , og en afstand mellem punkterne, på σ . dx og dy foldes med en cirkulær Gausskerne, med samme størrelse som Haar-wavelet responsene, og $\sigma_{Gauss} = 2.5\sigma$.



Figur 27:

Et sliding vindue på 60° , summerer alle vektorene, der ligger indenfor dets rækkevidde. Den længste vektor, udgør retningen, som beregnes med *atan2* (invers tangens, med 2 inputs). Dette er illustreret på figur 27. Der er her taget en beslutning om, at slidingvinduet udregner summerne af vektorer, indenfor seks positioner ($0^\circ - 60^\circ$, $60^\circ - 120^\circ$.. $300^\circ - 360^\circ$).

SURF deskriptoren samler data omkring interessepunktet, med et dataindsamlingsvindue der har størrelse 20σ . Vinduet skal orienters, ifht. den θ værdi, der er udregnet i detektoren. Billedet vendes, ved, at vende hele billedet, ved brug af ligning (40), og danne et integralbillede udfra det nye billede (Dette skridt er omkostningsfuldt, og der vil senere blive diskuteret optimeringer).

Dataindsamlingsvinduet er nu roteret og centreret omkring interessepunktet. Herefter skal vinduet deles op i 4×4 regioner, hver bestående af 5×5 punkter, der har ens afstand mellem sig - her er ikke blevet lavet subpixel-optimeringen, så hvis $\frac{20\sigma}{4}$ ikke er deleligt med 5, bliver der rundet ned.

Herefter skal Haar-wavelet responset udregnes, i x og y retningen (dx, dy , respektivt). Haar-wavelet skal udregnes for alle punkter i 5×5 felterne. Størrelsen på Haar-wavelet filteret, er 2σ . Disse værdier skal smoothes med et Gaussfilter, hvor $\sigma_{Gauss} = 3.3\sigma_{point}$.

For her af de 16 4×4 regioner, udregnes:

$$v_i = \sum dx, \sum dy, \sum |dx|, \sum |dy| \quad (46)$$

hvor v_i er beregningerne, for den i 'ende region. Der laves en vektor, hvor alle værdierne indgange i v sættes efter hinanden. Dette giver en $16 \cdot 4 = 64$ indgange stor vektor. Slutteligt, laves vektoren om til en enhedsvektor.

Algoritme

1. Orienteringen findes:
 - a) Orienteringen på punktet findes, ved at beregne dx, dy i et cirkulært område, med radius 6σ , og afstand σ , mellem punkterne.
 - b) dx, dy foldes med en Gauss kerne, med $\sigma_{Gauss} = 2.5\sigma$
 - c) Den længste vektor, af summerede vektorer, der ligger indenfor 60° , bruges til at tildele orientering θ til interessepunktet.
2. Et dataindsamlingsvindue roteret ifht. θ , beregnet som (40) med størrelse 20σ indsamler punkter, der har afstand σ mellem hinanden.

3. Haar-wavelet responset, dx , dy udregnes for de indsamlede punkter, med en filterstørrelse på 2σ , og dx, dy foldes med Gauss, hvor $\sigma_{Gauss} = 3.3\sigma$
4. For hver af de 4x4 regioner, udregnes (46). Dette sættes sammen til en vektor, der vil have 64 indgange

6.5 Matching

Matching foretages i to trin: Først bregnes den Euklidiske afstand som i (47), og derefter, sorteres dårlige matches fra:

$$||P|| = \sqrt{\sum_{i=1}^n (F_i - F'_i)^2} \quad (47)$$

Givet to sæt af punkter \mathbf{P}, \mathbf{P}' , hver bestående af (F, p) , udregnes der for hver feature i \mathbf{P} , afstandene til \mathbf{P}' . Den feature i \mathbf{P}' der ligger nærmest, bliver valgt og herved dannes et par af punkter, der tilføjes til et midlertidigt sæt, af korrespondancer. Når alle inputs er blevet matched, sorteres de efter afstand, med korrespondancer af korteste afstand først. Hvis $|\mathbf{P}| > |\mathbf{P}'|$, er det garanteret, at mindst ét element (og muligvis flere) fra \mathbf{P}' , har flere matches, fra \mathbf{P} . Dette er ikke hensigtsmæssigt, da det antyder, at minimum ét af korrespondancerne er forkerte. Der er derfor anvendt en metode, til at sortere dårlige korrespondancer fra.

En tilgang kunne være, udelukkende at kigge på de m bedste matches. Empirisk er det konkluderet, at denne tilgang virker med $m = 20$. Her fremkommer ingen outliers. Anvendes en større værdi for m bliver flere outliers tilføjet. Der skal derfor bruges en metode, der sortere outliers fra, og derved kunne konkludere, hvor mange inliers og outliers, de forskellige metoder resulterer i.

For at sortere forkerte korrespondancer fra, anvendes en statistisk udregning, der differentiere inliers og outliers.

Denne metode, fjerner korrespondancer, der har en euklidiske afstand, der er mindre end middelværdien, af de 10 bedste matches, multipliceret med 5, hvilket er en empirisk defineret konstant. Det resulterende sæt af punkters motion-vektor, skal nu udregnes: $MV = p_i - p'_i$. MV 's standard afvigelse σ beregnes nu i x og y retningen. For x kan dette skrives:

$$\sigma_x = \sqrt{\sum_{n=1}^N (x_i - \mu)^2} \quad (48)$$

Iterativt fjernes punkter, hvis afstanden fra middelværdispunktet, er større, end standardafvigelsesafstanden, her opskrevet som en indikatorfunktion, der determinere om en korrespondance korrekt:

$$indikator = \begin{cases} 1, & \text{hvis } \sqrt{(x_i - \mu_x)^2 + (y_i - \mu_y)^2} < \sqrt{\sigma_x^2 + \sigma_y^2} \\ 0, & \text{ellers} \end{cases} \quad (49)$$

Ovenstående bliver gentaget, indtil en grænseværdi for standardafvigelsen er nået. Processen er illustreret, i figur (49). Her ses, hvordan hver iteration fjerner forkerte korrespondancer, og bevæger sig tættere på en klynge af punkter, der må formodes at være inliers.

Algoritme

1. For hvert element i \mathbf{P} , udregnes den Euklidiske afstand som i (47), og den indgang i \mathbf{P}' , der har mindst afstand, bliver matched med \mathbf{P} . Denne liste kaldes M

2. M sorteres, så bedste match er først.
3. Motion-vektorene udregnes, sammen med (48) for x og y .
4. Indikatorfunktionen (49) bruges til at fjerne punkter, der er større end længden af standardafvigelsen
5. 4 gentages, indtil (48) når en grænseværdi

Litteratur

- [1] Droner kan mindske brug af pesticider, 2013. Københavns Universitet
[http : //www.science.ku.dk/presse/nyhedsarkiv/2013/droner _diku/](http://www.science.ku.dk/presse/nyhedsarkiv/2013/droner_diku/).
- [2] H. Aanæs, A. L. Dahl, and K. S. Pedersen. *On Recall Rate of Interest Point Detectors*. Electronic Proceedings of 3DPVT'10 : The Fifth International Symposium on 3D Data Processing, Visualization and Transmission, 2010.
- [3] H. Bay, T. Tuytelaars, and L. Van Gool. *SURF: Speeded Up Robust Features*. Katholieke Universiteit Leuven, 2006.
- [4] M. Brown, R. Szeliski, and S. Winder. *Multi-Image Matching using Multi-Scale Oriented patches*. Microsoft Corporation, 2005.
- [5] C. Harris and M. Stephens. *A Combined Corner And Edge Detector*. Plessey Research Roke Manor, United Kingdom, 1988.
- [6] T. Lindeberg. *Detecting Salient Blob-Like Image Structures and Their Scales with a Scale-Space Primal Sketch - A Method for Focus-of-Attention*. Computational Vision and Active Perception Laboratory (CVAP), Department of Numerical Analysis and Computing Science, KTH (Royal Institute of Technology), 1993.
- [7] T. Lindeberg. *Feature Detection with Automatic Scale Selection*. Computational Vision and Active Perception Laboratory (CVAP), Department of Numerical Analysis and Computing Science, KTH (Royal Institute of Technology), 1998.
- [8] D. G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints*. University of British Columbia, 2004.
- [9] K. Mikolajczyk and C. Schmid. *A Performance Evaluation of Local Descriptors*. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2005.
- [10] H. P. Moravec. *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. Robotics Institute Carnegie-Mellon University, 1980.
- [11] A. S. Ogale and Y. Aloimonos. *Invariant Features from Interest Point Groups*. Springer, 2008.
- [12] A. S. Ogale and Y. Aloimonos. *Shape and the stereo correspondence problem*. Springer, 2008.
- [13] K. M. Tinne Tuytelaars. *Local Invariant Feature Detectors- A Survey*. Springer, 2008.

A Appendix test

appendix test